GET: Получение всех user ID login пользователей

Severity **Priority Behavior Type** O Normal Positive **Functional** ↑ High Layer Is Flaky Milestone **Automation** Manual Not set No Status Is Muted Actual No

Pre-conditions

Пользователь зарегистрирован или создан. В коллекции DemoShopping создана переменная {{base_url}}

Parameters

операционная система: macOS Sonoma Версия 14.5 программное обеспечение: Postman 11.1.14 {{base url}}: https://qa.demoshopping.ru

Step	1
Action	Отправить GET запрос на {{base_url}}/users
Input data	
Expected result	Запрос успешно отправлен на сервер
Step	2
Action	Проверить код состояния
Input data	
Expected result	HTTP Status: 200 OK
Step	3
Action	Проверить время ответа
Input data	
Expected result	Время ответа меньше 999ms
Step	4
Action	Найти при помощи поиска нужного пользователя используя login
Input data	"user_id": 419, "login": "rimmaTest1"

Expected result

Пользователь успешно найден.



POST: Добавление нового продукта

Severity	Priority	Behavior	Туре
O Normal	↑ High	Positive	Functional
Layer	Is Flaky	Milestone	Automation
Not set	No	-	Manual
Status	Is Muted		
Actual	No		

Pre-conditions

В коллекции DemoShopping создана переменная {{base_url}}

B Pre-request "POST ${\{base_url\}}/products/id/{\{product_id\}\}}$ " создан скрипт, создающий новый продукт для последующего использования, идентификатор которого сохраняется в переменную $\{\{product_id\}\}$

Content-Type application/json

Значения ключей в теле запроса рандомизированы

Parameters

```
операционная система: macOS Sonoma Версия 14.5 программное обеспечение: Postman 11.1.14 {{base_url}}: https://qa.demoshopping.ru {{product_id}}: Динамическая переменная, заполняется автоматически
```

Step	1
Action	Отправить POST запрос {{base_url}}/add-product
Input data	Тело запроса в формате raw, JSON: { "name": "{{\$randomProductName}}", "description": "{{\$randomLoremSentences}}", "price": {{randomPrice}}, "category": "{{\$randomLoremWord}}", "manufacturer": "{{\$randomLoremWord}}", "imageUrl": "{{\$randomLoremWord}}, "freeShipping": {{\$randomBoolean}} }
Expected result	Запрос успешно отправлен на сервер
Step	2
Action	Проверить код состояния
Input data	
Expected result	HTTP Status: 200 OK
Step	3
Action	Проверить время ответа

Input data Expected result Step 4 Action Проверить тело ответа Input data Expected result Tело ответа содержит сообщение: "Продукт успешно добавлен с ID: "ID нового продукта"

РАТСН: Обновление цены и имени товара по ID

Severity	Priority	Behavior	Туре
O Normal	↑ High	Positive	Functional
Layer	Is Flaky	Milestone	Automation
Not set	No	-	Manual
Status	Is Muted		
Actual	No		

Pre-conditions

В коллекции DemoShopping создана переменная {{base_url}}

B Pre-request "PATCH {{base_url}}/products/id/{{product_id}}" создан скрипт, создающий новый продукт для последующего использования, идентификатор которого сохраняется в переменную {{product_id}}

Parameters

Step

4

операционная система: macOS Sonoma Версия 14.5 программное обеспечение: Postman 11.1.14 {{base_url}}: https://qa.demoshopping.ru {{product_id}}: Динамическая переменная, заполняется автоматически

Step	1
Action	Отправить PATCH запрос на {{base_url}}/products/id/{{product_id}}
Input data	Тело запроса в формате raw, JSON: { "name": "updated name from Rimma", "price": 159 }
Expected result	Запрос успешно отправлен на сервер
Step	2
Action	Проверить код состояния
Input data	
Expected result	HTTP Status: 200 OK
Step	3
Action	Проверить время ответа
Input data	
Expected result	Время ответа меньше 999ms

Action	Проверить тело ответа
Input data	
Expected result	Тело ответа содержит сообщение: "Товар частично обновлён"

DELETE: Удаление товара по ID

Severity **Priority Behavior Type** O Normal ↑ High Positive **Functional** Milestone **Automation** Is Flaky Layer Manual Not set No Status Is Muted

Actual No

Pre-conditions

В коллекции DemoShopping создана переменная {{base_url}}

B Pre-request "DELETE {{base_url}}/products/id/{{product_id}}" создан скрипт, создающий новый продукт для последующего удаления, идентификатор которого сохраняется в переменную {{product_id}}

Parameters

программное обеспечение: Postman 11.1.14

операционная система: macOS Sonoma Версия 14.5

{{base url}}: https://qa.demoshopping.ru

{{product_id}}: Динамическая переменная, заполняется автоматически

Steps to reproduce

Step 1

Action Отправить DELETE запрос на {{base url}}/products/id/{{product id}}

Input data

Expected result Запрос успешно отправлен на сервер

Step 2

Action Проверить код состояния, и ID, под которым был создан товар для удаления

Input data

Expected result HTTP Status: 200 OK



Step 3

Action Проверить время ответа

Input data

Expected result	Время ответа меньше 999ms
Step	4
Action	Проверить тело ответа
Input data	
Expected result	Тело ответа содержит сообщение: "Товар удалён"
Step	5
Action	Отправить GET запрос на {{base_url}}/products/id/:product_id и убедиться, ранее удаленного ID товара нет в списке
Input data	18009
Expected result	Удаленный товар в списке не найден. Скрин результата приложен
	### Committeeping (Frendam) Process recognised D ### OF

PUT: Полное обновление товара по ID

Severity	Priority	Behavior	Туре
O Normal	↑ High	Positive	Functional
Layer	Is Flaky	Milestone	Automation
Not set	No	-	Manual
Status	Is Muted		
Actual	No		

Pre-conditions

В коллекции DemoShopping создана переменная {{base_url}}

B Pre-request "PUT {{base_url}}/products/id/{{product_id}}" создан скрипт, создающий новый продукт для последующего использования, идентификатор которого сохраняется в переменную {{product_id}}

Content-Type application/json

Parameters

```
операционная система: macOS Sonoma Версия 14.5 программное обеспечение: Postman 11.1.14 {{base_url}}: https://qa.demoshopping.ru {{product_id}}: Динамическая переменная, заполняется автоматически
```

Step	1
Action	Отправить PUT запрос на {{base_url}}/products/id/{{product_id}}
Input data	Тело запроса в формате raw, JSON: { "name": "Example Product from Rimma nx2", "description": "This is an example product.", "price": 10000, "category": "Electronics", "manufacturer": "Example Manufacturer", "imageUrl": "http://example.com/image.jpg", "freeShipping": true }
Expected result	Запрос успешно отправлен на сервер
Step	2
Action	Проверить код состояния
Input data	
Expected result	HTTP Status: 200 OK
Step	3
Action	Проверить время ответа
Input data	

Expected result	Время ответа меньше 999ms
Step	4
Action	Проверить тело ответа
Input data	
Expected result	Тело ответа содержит сообщение: "Товар обновлён"